REMARKS

Entry of the foregoing amendments, reconsideration and allowance of the above-identified application are respectfully requested. Claims 1-19, 21-39 and 42-55 are currently pending. Claims 1, 22, 29, 37, 39, 42-45 and 47 have been amended. No new matter has been added.

Regarding entry of the foregoing amendments, it is respectfully submitted that the amendments to the claims are presented solely to address the formal rejections under 35 U.S.C. §112, first and second paragraphs, as described in detail below. No new issues are raised by these amendments, nor would a further search be required by entry of these amendments. Accordingly, entry of these amendments is respectfully submitted to be appropriate notwithstanding that this application is currently under Final Rejection, as they will serve to simplify the issues under consideration. Regardless of whether these amendments are entered, or not, at this junction of the prosecution, the Remarks below regarding the rejections pursuant to 35 U.S.C. § 103 are pertinent to the claims in both their amended and unamended forms and, therefore, consideration thereof is respectfully requested.

Claims 1-19, 21-39, and 42-55 stand rejected under 35 U.S.C. § 112, first paragraph, as allegedly failing to comply with the written description requirement. More specifically, the Official Action states that the “claim(s) contains subject matter which was not described in the specification in such a way as to enable one skilled in the relevant art that the inventor(s), at the time the application was filed, had possession of the claimed invention”. However, this rejection is respectfully traversed because the subject matter in both the unamended and the amended

claims 1-19, 21-39, and 42-55 is respectfully submitted to be supported in the original specification for at least the following reasons.

As regards independent claims 1, 22, 42-45 and 47, the Official Action points to an alleged lack of supporting disclosure for the feature that the packet records are stored and arranged for output in an exit order and for the feature that the packets or data portions are output in accordance with packet records being output from the queuing means. Consider, however, the following information provided in Applicant's originally filed specification.

The opening pages of the specification set out the problems with prior art solutions to dealing with data packet sorting or scheduling. Normally, a "queue first, think later" approach is adopted but this leads to the problems recognized in the specification. The exemplary embodiments use the reverse philosophy by assigning an exit order to incoming packets in real time **before** any other packet processing is carried out, especially any packet re-ordering and/or packet record re-ordering.

Referring particularly to Figure 2 and to the corresponding pages 6-7 of the original specification, packet record portions are generated from packet headers and are assigned an exit order by means of a processor (22). In practice, the processor can be a parallel processor, which offers immense processing power and speed to achieve this in real time. At this point, nothing has been done to affect the order of the packets in relation to their arrival order. The processor (22) simply calculates the order in which the data portions are to be output. The record portions themselves, however, do not have to be placed in that order when they are output from the processor.

As described in page 6 of Applicant's specification, these packet records are then processed in an orderlist manager (24). This is an intelligent queuing system that decants the packet records into bins, possibly over a number of iterations, thereby forming an exit order queue. The data portions (payload) of the packets pass to a memory hub (21), where they are held in no particular order. They remain there until called upon by the output block (25). Page 7, third paragraph, states that the output (25), "in dependence on the exit order queue held in the Orderlist Manager 24, instructs the Memory Hub 21 to read out the corresponding packets in that required order". The fact that the packet records are held in a managed queue where the packet records are arranged in ***exit*** order, tells the person of ordinary skill in the art that the records are output, otherwise there would be no need for a queue. Bearing in mind the fact that the original claims and the statements in page 4 (eg lines 19-22) comprehended two options, where (i) whole packets are stored in the memory hub or (ii) only data portions are stored in the memory hub, the output (25) instructs the memory hub (21) to read out the corresponding packets or the data portions of the packets in the originally assigned exit order.

For at least the foregoing reasons, it is respectfully submitted that the originally filed specification gave the person of ordinary skill in the art sufficient and enabling instructions as to how to effect the assignment of an exit order to incoming packet records, to process the packet records to form an exit-order queue, to store packets or data portions of packets in a memory, and to read out the stored packets or data portions in dependence on the exit order queue. This system is expressed in

claim 1 as currently amended, in which the claimed system provides for assignment means (e.g., the processor 22) operable only on packet records to assign an exit order to the packets in real time; queue means (e.g., the orderlist manager 24) responsive to the assignment means to store and arrange the packet records for output in the exit order; and a memory means (e.g., the memory hub 21) for storing packets (as per option (i) above) or packet data portions (as per option (ii) above); and the packets or data portions, as the case may be, are output from the memory means in dependence on the exit order queue held in the queue means (e.g., the orderlist manager, as per page 7 of Applicant’s specification) for the corresponding packet records. In this way, the packet is output in the exit order originally assigned to the packet header at the input.

However, in order to remove any doubt as to whether the record portions are actually output, which is clearly the case from the above analysis and the originally filed specification, claim 1 is proposed to be amended above to remove any reliance on the packet records being “output”. Specifically, it is proposed to amend claim 1 to specify “queue means responsive to said assignment means for storing and arranging said packet records in said exit order” (i.e., to delete “for output”) and so that the final subparagraph states “said packets or data portions being output from the memory means in accordance with the exit order of the corresponding packet records in the queue means”. This expression is supported by at least page 7, third paragraph, of the originally filed specification.

For at least the foregoing reasons, it is respectfully requested that that this amendment be entered for claim 1 and that the corresponding amendments for each

of independent claims 22, 42, 43-45 and 47 likewise be entered. Then, reconsideration and withdrawal of the rejection of claim 1-19, 21-39, and 42-55 under 35 U.S.C. § 112, first paragraph, are also respectfully requested.

Claims 37-39 stand rejected under 35 U.S.C. § 112, second paragraph, as allegedly being indefinable for failing to particularly point out and distinctly claim the subject matter which Applicant regards as the invention. Regarding claims 37-39 the Official Action states that there is “no antecedent basis for ‘said processor elements’”. Accordingly, it is proposed above in claim 37 to replace the passage “wherein said tables are stored locally to said processor elements” to read “wherein said tables are stored locally to processor elements of said processor”. Entry of this amendment, reconsideration and withdrawal of the rejection of claims 37-39 under 35 U.S.C. § 112, second paragraph, are respectfully requested.

Claims 1, 2, 5, 8, 11, 22, 23, 26, 27, 32, 42-45 and 47 were rejected under 35 U.S.C. §103(a) as allegedly being unpatentable over Sindhu et al (USP 7,342,887) in view of Brunheroto et al. (USP 6,643,298). However, this ground of rejection is respectfully traversed for at least the following reasons.

Sindhu describes a switch interconnecting input and output line cards in a network. The switch operates by sending a request to an output/destination line card for a packet or a portion (cell) of a packet to be sent to it through a switch fabric from an input/source line card. The packet header contains information identifying the source line card and the destination line card. If the destination line card is able to accept the request, it issues a grant signal that passes back through the switch fabric to the requesting source line card. The grant signal contains only source and

destination line card numbers (see, e.g., column 10, lines 48-50 of Sindhu). A packet or cell is then sent through the switch fabric to the destination line card. Where several cells make up a packet, all of the cells of that packet are sent in this way and reassembled into a packet at the destination line card.

Significantly, the grant signal described in Sindhu does not relate to a particular data cell. This is especially true of the cell that led to the request. Rather, it is an indication that there is enough bandwidth available for transmissions to take place between the particular source line card and the designated destination line card. To be more exact, it is a general authorisation from a particular destination line card to a requesting source line card to transmit a data cell in the next time slot. Thus, where data cells of high priority are put at the head of a queue, the grant signal would allow such a high priority cell to be transferred through the switch fabric without a specific authorisation. In effect it rides on the back of a grant signal generated in respect of another request. There is therefore no correspondence between the request and the transmitted data cell in the system of Sindhu.

The passage of Sindhu disposed between column 7, line 55 and column 8, line 15 identified in the Official Action describes a combination source/destination line card illustrated in Figures 3 and 5. A network interface logic Nw (70) divides the packets into cells and writes them to memory system 76. The logic Nw also extracts keys from the packet headers and sends them to a route lookup system R (74) to determine the appropriate destination line card. A fabric interface logic Nf (72) reads cell data from memory system 76 and forwards one “data transfer unit” (i.e., a cell that includes cell data plus request and grant information - Column 6, lines 46-47) at

a time to the switch fabric for the appropriate destination line card. The Nf logic is also responsible for receiving requests, grants and data cells from the switch fabric.

Column 9, line 10 to Column 10, line 11 of Sindhu continue on to explain with reference to Figures 6 and 7 that a request generator 78 stores packet headers in N header queues, one per destination line card. The headers are transferred to the request generator from the memory system or are retrieved from the memory system by the request generator. The packet headers contain information as to the source, destination and length of the packet. The request generator generates a number of requests, dependent on the size of the packet. The requests are transmitted via logic 84 Figure 7 to the first stage F1 switch (Figure 3) associated with the particular line card. When the requests are acknowledged via third stage F3 switch, a grant generator 80 Figure 6 generates grant signals which are sent to the F1 switch associated with the destination line card.

A data cell transmitter 82 of the fabric interface logic Nf also contains a number N of packet header queues 104 Figure 9, one per destination line card. The request generator 78 Figure 6 sends the headers to the queues 104 when it selects a packet to transmit from its header queues 88 Figure 7. A fetch block 102 Figure 9 receives grant signals returned by the destination line cards. For each such grant signal, the fetch block identifies an appropriate header queue104 to obtain the address of the next data cell stored in memory system 76 to be transferred to the designated destination line card. The fetch block retrieves that designated cell from the memory system and forwards it to the switch fabric stage F1 or instructs the memory system to transfer it direct.

However, Sindhu deals with an entire router/switch implementation and neatly encapsulates the distinction between two fundamental forms of router/switch implementations, as follows:

In Column 1, lines 40-50, Sindhu states:

"Two conventional approaches to designing a router include a central memory approach and a central switch approach. In the central memory approach, all of the buffering for the router is provided by a single logically-centralized memory buffer with little or none provided by the individual line cards. In the central switch approach, input and output line cards are coupled by a switch fabric where each line card provides delay-bandwidth buffering. The switch fabric typically includes no buffering. The central memory approach is lower cost, but the central switch approach is easier to scale to greater numbers of line cards."

In Column 2, lines 4-9, Sindhu further states:

"A centralized controller works well for a relatively small router, however it rapidly becomes unimplementable with increases in the number of line cards. The storage and processing requirements of the central controller grow at least as the square of the number of line cards, making this approach of limited utility in scaling routers to a large size".

These statements of Sindhu convincingly underline the reason why Applicant's solution to these problems provides a novel and inventive approach to the centralized controller format of router/switch.

More specifically, Sindhu provides a clear teaching and underlying philosophy relating to the use of multiple output queues (see for example at least Column 10, lines 18-20 of Sindhu). In essence, Sindhu describes a distributed set of output queues (i.e., a plurality of queues as criticized in the quoted statements above) so, if there were say 10 line cards, each line card would hold 10 queues. In total there

would be 100 such queues which the controller and the destination line card would have to arbitrate between as regards the requests being made of it by all the source line cards.

In contrast, the implementation of the centralized processing method that Applicant describes maintains a single output queue (see e.g., claim 6 and page 7 of the description). The significance of Applicant's approach is the exploitation of the high memory bandwidth of the processing elements of the processor employed to process the packets and by doing so overcome the issues identified by Sindhu associated with the central memory architecture whilst providing a high data-rate architecture. In doing so, Applicant is able to remove (a) the need to output queue at the input line cards, (b) the need to police the switch bandwidth by using a grant & request format, (c) the need to use fixed-size cells (as taught by Sindhu as regards dividing a packet into one or more cells), (d) the need to spray cells across the switch fabric, and (e) the need to reassemble cells that have been sprayed to reform packets before ultimately transmitting those cells from the destination port. All of these features are necessary, as Sindhu explains, in order to achieve a non-blocking and fair switching operation for a central switch based architecture.

The ability to handle high-priority traffic (within a line-card) relies on higher-priority traffic employing grants generated by previous requests, because requests and grants are not tightly coupled to specific transfers. The ability to prioritize between line cards relies on the controller. Within the central memory approach the processing elements have full control over how the packets are ultimately prioritized.

Even to the extent that one of ordinary skill in the art could view the output

queuing that is performed on the output of each source line card in Sindhu as being similar to the output queuing claimed in claim 1, this would not have motivated such a person to have reached Applicant's claim 1 combination. Consider that, rather than generating a plurality of distributed output queues from which to select and switch packets to the ultimate destination, Applicant implements, through the use of the processing elements of the parallel processor, a single output queue containing packet records assigned with an exit order. The single output queue is processed further in the orderlist manager, where the queue is stored and arranged such that the packet records are placed in exit order.

Claim 1 states unequivocally that the queue means stores and arranges the packet records in the assigned exit order. This is not the case with Sindhu, as has now been extensively argued above. Rather, Sindhu delivers a plurality of queues, each of which contains some of the packet headers that constitute the overall output queue and whose final exit order is yet to be determined following final arbitration through the switch fabric.

For at least this reason and for the other reasons set out above, it is respectfully submitted that one of ordinary skill in the art would not have regarded Sindhu as teaching at least this feature of Applicant's claim 1 combination. Moreover, there would have been no motivation for the skilled person to have considered combining Sindhu with Brunheroto at least in any manner which would have arrived at Applicant's claim 1 combination. Similar comments apply to the other independent claims.

Accordingly, reconsideration and withdrawal of this ground of rejection are

respectfully requested.

Claims 3, 17-19, 24, and 37-39 are rejected under 35 U.S.C. §103(a) as unpatentable over Sindhu et al. (USP 7,342,887), Brunheroto et al. (USP 6,643,298), and De Silva et al. (USP 7,499,456). Claims 4, 21 and 25 were rejected under 35 U.S.C. §103(a) as unpatentable over Sindhu et al. (USP 7,342,887), Brunheroto et al. (USP 6,643,298), and Yazaki et al. (US Publ. No. 2005/0163049). Claims 6-7, 28 and 29 were rejected under 35 U.S.C. §103(a) as unpatentable over Sindhu et al. (USP 7,342,887), Brunheroto et al. (USP 6,643,298), and Kiremdjian et al. (US Publ. No. 2003/0081623). Claims 9-10, 30, and 31 were rejected under 35 U.S.C. §103(a) as unpatentable over Sindhu et al. (USP 7,342,887), Brunheroto et al. (USP 6,643,298), and Donis et al. (US Publ. No. 2002/0075882). Claims 12-16, 33-36 and 46 were rejected under 35 U.S.C. §103(a) as unpatentable over Sindhu et al. (USP 7,342,887), Brunheroto et al. (USP 6,643,298) and Wilkinson et al. (USP 6,094,715). Claims 48-55 were rejected under 35 U.S.C. §103(a) as unpatentable over Sindhu et al. (USP 7,342,887), Brunheroto et al. (USP 6,643,298), Wilkinson et al. (USP 6,094,715), and De Silva et al. (USP 7,499,456).

All of these grounds of rejection are predicated on the Sindhu patent as the primary reference. However since the other documents cited in these various grounds of rejection fail to remedy the above-described deficiencies of Sindhu, it is respectfully submitted that none of these other combinations of documents would have motivated one of ordinary skill in the art to have reached these dependent claims. Accordingly, reconsideration and withdrawal of these grounds of rejection are also requested.

All of the objections and rejections raised in the Office Action having been addressed, it is respectfully submitted that this application is in condition for allowance and a notice to that effect is earnestly solicited. Should the Examiner have any questions regarding this response or the application in general, she or he is invited to contact the undersigned at (540) 361-1863 to expedite prosecution of this application.

Respectfully submitted,

POTOMAC PATENT GROUP PLLC

By: /stevenmdubois/
Steven M. duBois
Registration No. 35,023

Date: July 17, 2009

Potomac Patent Group, PLLC
P.O. Box 270
Fredericksburg, VA 22404
(540) 361-1863